1.开源开发有哪些人员角色

## 贡献者

“贡献者”是指为 AOSP 源代码做贡献的人。贡献者可以是 Google 或其他公司的员工，也可以是不属于任何公司的个人开发者。贡献者之间并无差别，他们都使用相同的工具（Git、Repo 和 Gerrit），遵循相同的代码审核流程，并遵守相同的代码样式要求，等等。

**开发者**

“开发者”是指编写在 Android 设备上所运行应用的工程师。开发者和贡献者通常具有相似的技能组合，但开发者是在使用平台，而非对平台做出贡献，因此 AOSP 将开发者视为客户。我们经常会提到开发者，尽管从技术层面来讲，其并不是 AOSP 中的一个单独角色。

**验证者**

“验证者”可以测试更改请求。当有人向该项目提交了大量高质量代码后，项目负责人可能会邀请他们成为验证者。

**注意**：目前，验证者与审批者类似。

**审批者**

“审批者”是经验丰富的 AOSP 成员，他们在技术和设计方面为此项目做出了重大贡献。在代码审核流程中，审批者会决定是纳入还是排除某项更改。项目负责人（通常是 Google 员工）负责选择审批者，有时也会将在特定项目中展现出杰出专业技能的验证者晋升为负责人。

2. 创建和开展开源项目需要注意的问题

## 开源协议

相信很多小伙伴在开发的时候都默认遵循 **不重复造轮子(偷懒)** 这一原则，只要有了思路就马上在GitHub搜索一下，看看是否有人已经做了，如果已经有做好的，自然就不客气啦，拿过来修改一下就能用，不由得心中暗喜，又省了好多时间能用来(LOL)。然而你可能没注意到，在诸多的开源代码中存在一些陷阱(约束)，就是开源协议。

## 为什么要添加开源协议?

首先是对作者的保护，防止知识成果被恶意利用。

开源协议中一般都包含免责声明(禁止代码的作者承担代码使用后的风险及产生的后果)，比如你开源了一个破解智能锁的代码，如果有人利用这个去盗窃导致他人损失，你是无需承担责任的。

其次是对使用者的保护，方便使用者。

使用者一看就知道自己允许进行哪些操作，不允许进行哪些操作。

未添加协议的代码默认是作者保留所有权利的(*对此不同国家的法律可能稍微存在区别*)，这就像一颗定时炸弹，如果你在项目中使用了这一份没有协议的代码，原作者只要能证明你未经许可使用了他的代码，是能够起诉你的。

**确保项目代码始终可构建**

项目代码包含构建脚本和说明。使其尽可能容易地构建成功和测试项目代码。

**与用户社区联系**

用户社区是使用开源项目产品的一群人。用户社区很少直接向项目代码贡献任何东西，但却往往会提出很多问题。确保这些问题得到回答。健康的用户社区可以为采用者社区提供信息。

**与开发社区联系**

开发社区由项目团队成员和贡献者组成。为社区内的沟通提供众所周知的渠道。明确的沟通渠道将有助于开发人员的合作。

**有计划**

很容易就会陷入只专注软件开发的模式，但和任何进程(特别是软件开发过程)一样，有些极致的方法至关重要。确保你的项目采用开发方法，并且有人拥有该流程(例如项目负责人)。制定计划可以帮助开发人员知道他们在哪里可以贡献最大的价值，并使采纳者和扩展者更容易实现他们自己的计划(从而取得成功)。像对待任何其他软件开发项目一样地对待你的开源项目

3. 开源开发为什么要进行版本控制，如何进行版本控制？

原因：因为版本控制可以保存了数据当前状态以及之前每一个提交的历史状态，可以回退到任意一个版本节点；在保存每一个版本的文件信息时不重复保存数据，节约存储空间，提高运行效率；可以清楚到看到不同版本间修改的内容；可以多人协作，团队开发；

方法： 使用Git进行版本控制。

4. GitHub地址

https://github.com/zwk-zwk